**Amendment to the Specification:**

**On page 15, please replace the abstract with the following rewritten paragraph:**

## ABSTRACT OF THE DISCLOSURE

Improved techniques for representation of objects in a Java programming environment are disclosed. The techniques are highly suitable for representation of Java objects inside virtual machines, especially those that operate with limited resources (e.g., embedded systems. A cluster of Java object representations is disclosed. Each of the Java object representations provide a reference to a Java object and a reference to the class associated with the Java object. Accordingly, a two-tier representation is provided which allows efficient implementation of applications which need to access information regarding both Java objects and classes. As a result, quick access to information regarding Java objects can be achieved.

**Please replace the paragraph [0033] at page 9 with the following rewritten paragraph:**

[0033] It should be noted that the internal object representation 300 may include an identifier that uniquely identifies the Java object. As will be appreciated by those skilled in the art, the identifier can be a hash key. In one embodiment, the address of the first reference 302 is used as the hash key. It should also be noted that the first and second references ~~232~~ **302** and 310 represent two consecutive memory addresses. As such, each of the first and second references 302 and 310 can be four consecutive bytes (one word) in a memory portion of the virtual machine.

**Please replace the paragraph [0035] starting at page with the following rewritten paragraph:**

[0035] Initially, at operation 402, a sequential read of a cluster of two-tier Java object representations is initiated. Next, at operation 404, a determination is made as to whether Java objects or Java classes are to be identified. If it is determined at operation 404 that Java objects are to be identified, the method 400 proceeds to operation 406 where references to Java objects are sequentially read from the cluster of two-tier Java object representations. Thereafter, at operation ~~410~~ **408**, the memory addresses that have been read are marked. The method 400 ends following operation

408. However, if it is determined at operation 404 that Java objects are to be identified, the method 400 proceeds to operation 410 where references to Java classes are sequentially read from the cluster of two-tier Java object representations. Thereafter, at operation 408, the memory addresses that have been read are marked. The method 400 ends following operation 408.